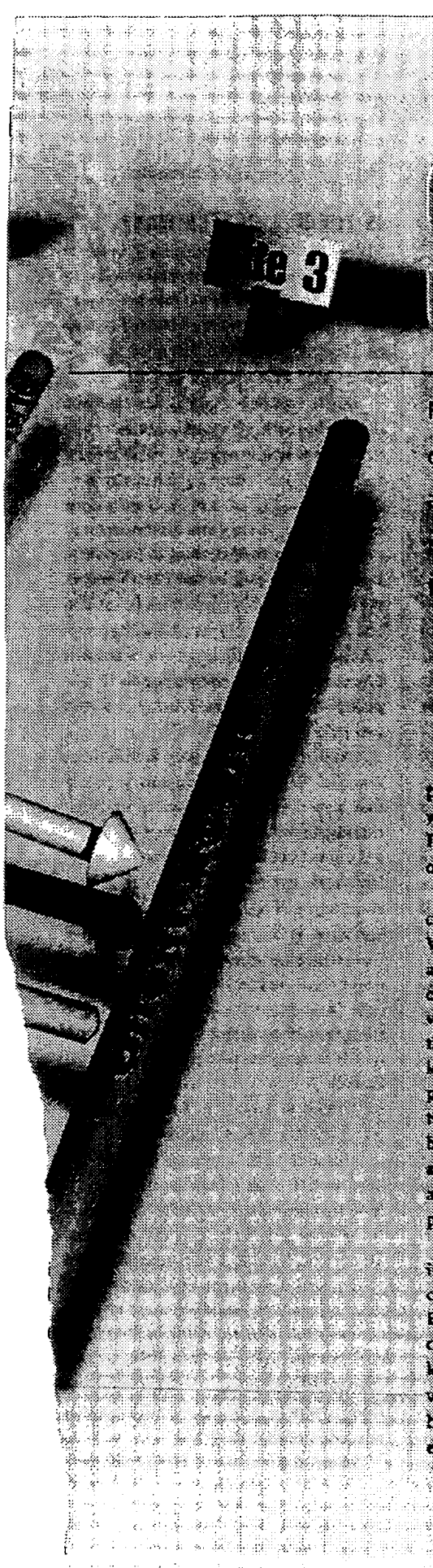


SIP Rules!

Internet e-mail and the web are killer apps par excellence: elegant, open, debuggable, extensible – based on simple protocols and standards, and implemented across a cloud of shared, multipurpose servers and resources. What if IP telephony looked like this? That's SIP's vision. Some say it's telephony's future.

As sea as IP telephony takes off, its proponents still make reassuring noises about legacy interoperability. "Don't worry," they say. "Our gateways will ping right into the face of that old PBX — you won't even know they're there. Relax. IP telephony doesn't change everything overnight."

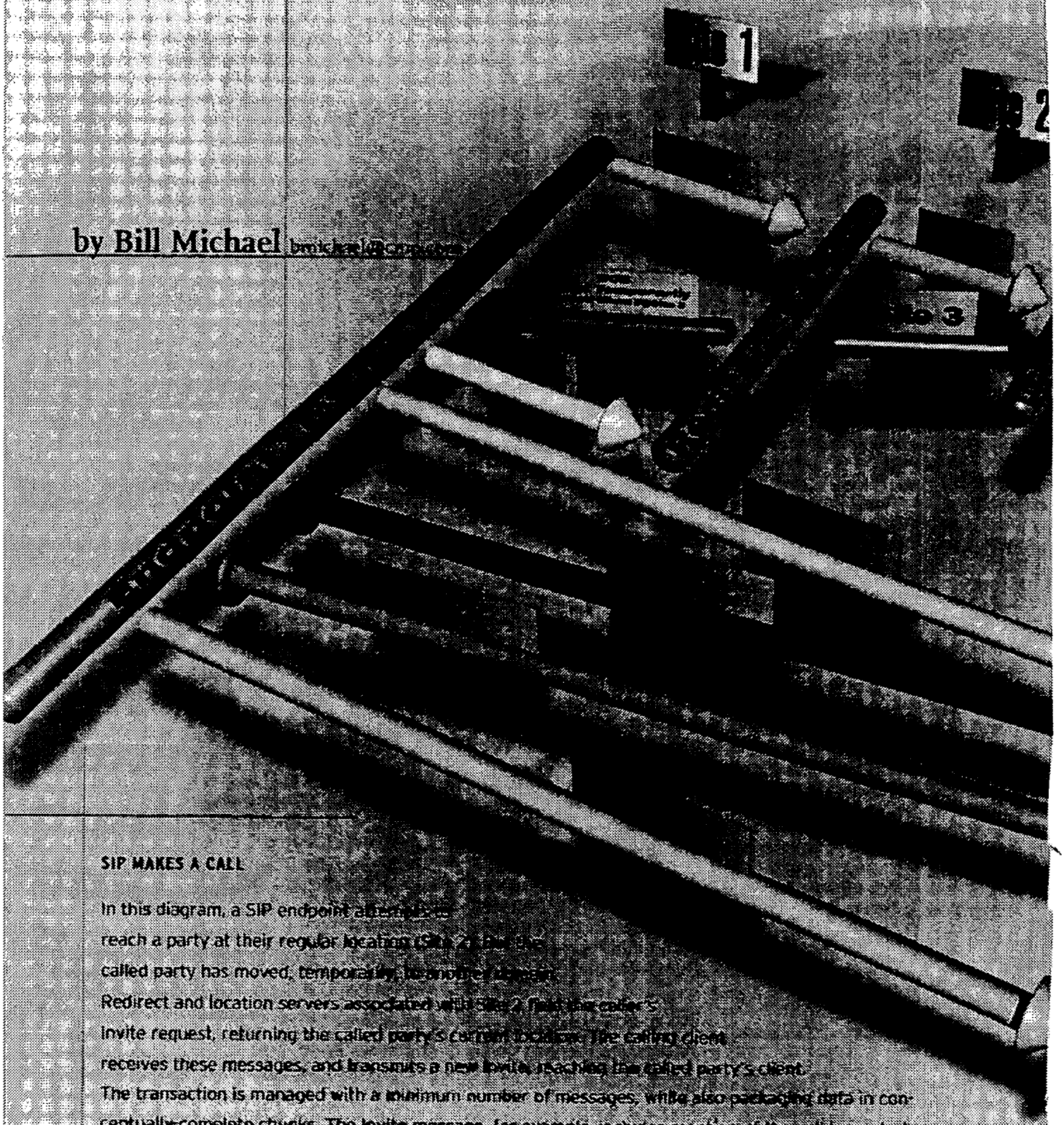
But it does. Or at least it would, if legacy telecom weren't hanging around VoIP's neck like an enormous practical and conceptual boat anchor. To achieve critical mass, today's IP telephony services and products must integrate with the PSTN and fit more or less neatly into legacy niches at the CO and customer premise. The result, perhaps inevitably, is that function tends to follow form: because these new networks, services, and devices interoperate with the older architecture, they tend willy-nilly to recapitulate its dominant themes.

Dominant themes of the old network include concentration, increased local complexity, verticality, determinism. Everywhere in the converging communications space, we see evidence of these legacy tendencies at work — often fusing with next-gen principles in strange hybrids of old and new. Right now, for example, VoIP network ASPs like Telera are working hard to concentrate gateway, call processing, and IVR functionality in boxes at the network edge. Considered one way, this is quite "next-gen" — Telera is putting all their legacy stuff (gateway components, conventional IVR) in one box, and pushing it out of the IP core. In another sense, it's quite "old network." In Telera's model, the gateway — conceptually a fairly simple device — becomes more complicated, more important, less generic.

Another example: IP PBXs, which we touted two issues back, are (naturally enough) being packaged as drop-in replacements for conventional phone systems. So products must either incorporate trunk- and station-side gateways, or use peripheral gateways for conventional connectivity. The gateways are conceptually disposable — most architectures handle call-direction and feature service entirely in software, on the IP side. But they add real complexity to what would otherwise be elegant, simple systems; becoming likely points of critical failure; and increasing cost. And until everything in the world goes IP (or at least until gateway service moves up into the network, where it belongs) they won't go away.

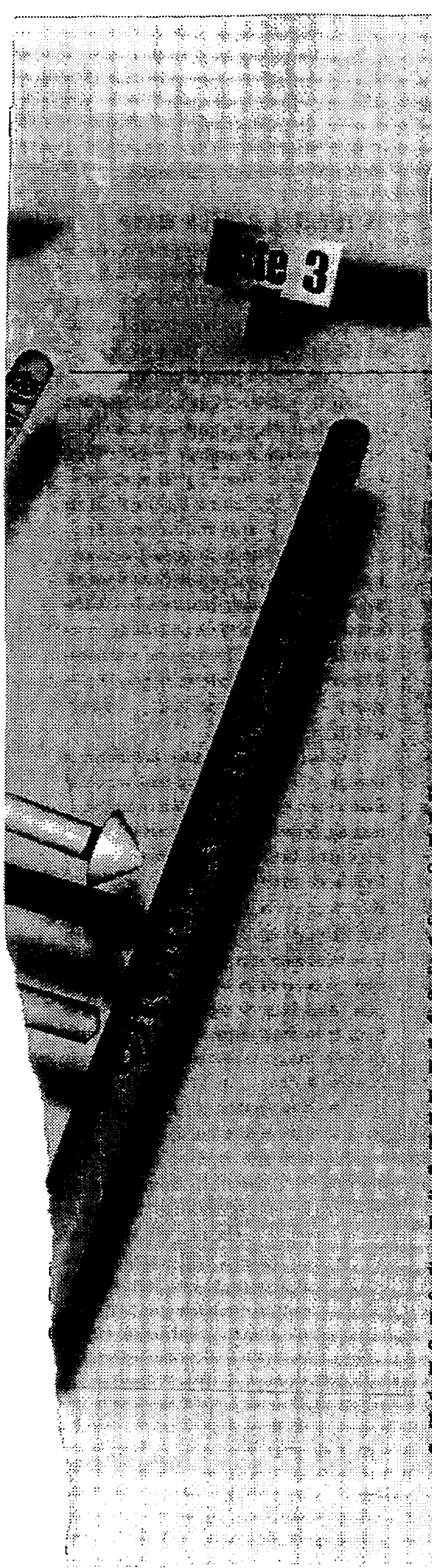
Centralized, deterministic, box-bound "telephony thinking" inflects most VoIP

by Bill Michael bmichael@



SIP MAKES A CALL

In this diagram, a SIP endpoint attempts to reach a party at their regular location (Call 2, then 3) as the called party has moved, temporarily, to another domain. Redirect and location servers associated with Site 2 field the caller's invite request, returning the called party's current location. The calling client receives these messages, and transmits a new invite reaching the called party's client. The transaction is managed with a minimum number of messages, while also packaging data in conceptually complete chunks. The invite message, for example, includes mention of the call type, designates preferred codecs and a realtime protocol. The OK message encapsulates similar information for the called party. Messages are formatted as human-readable text, based on HTTP 1.1 syntax.



SIP Rules!

Internet e-mail and the web are killer apps par excellence: elegant, open, debuggable, extensible – based on simple protocols and standards, and implemented across a cloud of shared, multipurpose servers and resources. What if IP telephony looked like this? That's SIP's vision. Some say it's telephony's future.

[redacted] as IP telephony takes off, [redacted] proponents still make reassuring noises about legacy [redacted] "Don't worry," they say. "Our gateways will plug right into the back of that old PBX — you won't even know they're there. Relax. IP telephony doesn't change everything overnight."

But it does. Or at least it would, if legacy telecom weren't hanging around VoIP's neck like an albatross: practical and conceptual baby-bathwater. To achieve critical mass, today's IP telephony services and products must integrate with the PSTN and fit more or less neatly into legacy niches at the CO and customer premise. The result, perhaps inevitably, is that function tends to follow form: because these new networks, services, and devices interoperate with the older architecture, they tend willy-nilly to recapitulate its dominant themes.

Dominant themes of the old network include concentration, increased local complexity, verticality, determinism. Everywhere in the converging communications space, we see evidence of these legacy tendencies at work — often fusing with next-gen principles in strange hybrids of old and new. Right now, for example, VoIP network ASPs like Telera are working hard to concentrate gateway, call processing, and IVR functionality in boxes at the network edge. Considered one way, this is quite "next-gen" — Telera is putting all their legacy stuff (gateway components, conventional IVR) in one box, and pushing it out of the IP core. In another sense, it's quite "old network." In Telera's model, the gateway — conceptually a fairly simple device — becomes more complicated, more important, less generic.

Another example: IP PBXs, which we touted two issues back, are (naturally enough) being packaged as drop-in replacements for conventional phone systems. So products must either incorporate trunk- and station-side gateways, or use peripheral gateways for conventional connectivity. The gateways are conceptually disposable — most architectures handle call direction and feature service entirely in software, on the IP side. But they add real complexity to what would otherwise be elegant, simple systems: becoming likely points of critical failure and increasing cost. And until everything in the world goes IP (or at least until gateway service moves up into the network, where it belongs) they won't go away.

Centralized, deterministic, box-bound "telephony thinking" inflects most VoIP

protocols, as well — but in a topsy-turvy way. When telecom-heads confront IP transport, the basic old telecom model ("drive dumb endpoints with concentrated, vertically-integrated intelligence") gets turned sideways. More intelligence is concentrated at the endpoints — both to manage what's assumed to be an extremely-fallible network, and to handle facilities negotiations between more- and less-complex devices, all residing somewhere on the (presumed) vertical continuum from voice, to video, to data, to all-of-the-above. But this distribution of intelligence to endpoints doesn't make things simpler at the network core.

H.323 — derived from the wireline videoconferencing protocol, H.320 — is an obvious case in point: complex, deterministic, vertical. The protocol — spread across at least six major documents (not counting optional addenda and semi-official commentary) — defines every component of a voice/video/data conferencing network: terminals, gateways, gatekeepers, MCUs, and other feature servers. H.323 uses ISDN-style Q.931 signaling for call setup, plus other protocols — RAS and H.245 — for terminal/gatekeeper negotiations and codec/facilities handshaking. All these protocols — dozens of back-and-forth messages — must be managed to set up a simple, point to point voice call.

H.323 is a fairly-stable standard — you can go to a range of third parties and buy stack components for host deployment, or terminal implementations. Interoperability tests are proceeding. Scaleability concerns are being addressed. H.323 gateway networks are deployed. H.323 PC clients are widely available — NetMeeting, for one, representing a kind of low-end, de-facto standard for client-side functionality. The first, relatively low-cost H.323 telephones are in the pipeline for second-quarter appearance. There's no question that H.323 works.

But anyone who looks at the standard should have questions. The New Network isn't going to be nearly as fallible as H.323 presupposes. (Nor is H.323 especially robust — the numerous messages required to set up calls mean plenty of targets for line-hits. Call setup failure due to packet loss is one of the things CT Labs measures, when they test H.323 gateways — performance of some systems is truly dismal.) There's going to be plenty of bandwidth — the idea that IP telephony is going to happen across 4.8 kbps compressed connections is pretty-well outmoded, as is the idea that most IP connections will have to negotiate bandwidth shifts, mid-call. Do we really need a facilities-negotiation sub-protocol (H.245) that not only manages mid-call codec changes, but is actually capable of (hold on to your hat) changing H.245 revision-levels, on the fly? And why would anyone want to use ISDN-style signaling to set up calls across an IP network?

Yes, H.323 works. But there's something fundamentally wrong-headed about it. It's all about concentration and control — dynamics diametrically opposed to the simple, open, horizontal, multi-purpose philosophy of pure Internet technologies like e-mail and the web.

This determinism — in combination with other "legacy" tendencies influencing the development of IP telephony networks and CPE — entails a fundamental risk to the converging communications economy. The IP telephony revolution could hang fire — or actually fail — if persistent legacy characteristics obscure real "killer app" opportunities, or hamper IP telephony's ability to elide with the Net's most powerful technologies.

MORE SIP TO COME!

We've ju[redacted] this issue – a[redacted]ting products and tools. Next issue, we'll be taking a much closer look: both at SIP internals and at some of the issues and challenges now facing the SIP community and the Net at large. Our June tutorial comes courtesy of consultant Richard Schockey, a member of the IETF and president of Shockey Consultants. Stay tuned.

IS THERE A BETTER WAY?

SIP — the session interface protocol — may offer a better way to do telephony in an IP environment. SIP comes at the challenge of converged communications from a horizontal, Net-head perspective. The result is a simple protocol with profound implications.

Like the web — originally designed as a document-sharing system for academics — SIP originated with a simple, practical brief. In the mid-90's, Henning Schulzrinne — now associate Professor in the Departments of Computer Science and Electrical Engineering at New York's Columbia University; Jonathan Rosenberg — now Chief Scientist at SIP software maker dynamicsoft; and several others began work on a signaling protocol that defines call setup and teardown functions as simple text commands. IP telephony — as we conceive it today — wasn't yet on their radar-screen.

"At that stage," Professor Schulzrinne remarks, "IP telephony as a term probably didn't even exist, at least not in my community. Initially, SIP was intended to create a mechanism for inviting people to large-scale, multipoint conferences. After a short while, it became clear that technology-wise, it was not a significant jump from where we were to setting up point-to-point conferences — essentially 'phone calls.' And once 'IP telephony' became the thing to do, then people started looking primarily at using the protocol for voice applications. But the emphasis of SIP has always been to remain as independent as possible of the media it underlies."

This abstractive approach is one of the keys to SIP's simplicity and elegance. Jonathan Rosenberg explains: "It's actually not even just media that SIP abstracts. The protocol makes a total separation between what it means to be a session, and what it means to establish one. SIP talks about establishing or modifying or terminating a session, but that particular session could just as easily be a multiplayer Doom game as it could be a voice channel or a videoconference."

So too, the decision to format SIP mes-

sages as text (instead of more bandwidth-economical, packet-size observant, theoretically 'easier to parse,' and of course, controllable binary) was a profound one. Text is human-readable. Text is flexible: Interpreting text demands some parsing intelligence — this renders applications more robust and lends itself to innovation.

Most to the point, text processing lies at the heart of the Net's true killer apps: e-mail and the web. Extensible tagging systems, document identification, data-type declaration, parsing methods and software for same — all these have been worked over, normalized, and shared-out by Net-heads in the process of bringing the modern Net online. Schulzrinne and his colleagues understood all this, and made a second leap: They decided SIP text messages would be composed in standard ISO UTF-8 (ASCII Unicode) characters, using HTTP 1.1 syntax.

A SIP message looks like the first five or six lines of source behind a well-formed web page (the part that says: 'Content-type: etc., etc.'). SIP messages look this way because that's what they are — an application of the Net's simplest, most widely-implemented, most general-purpose system for document-type declaration. SIP also adopts the conventional URL format for addressing server entities and people: Your SIP "phone number" will likely be 'yourname@yourhost.com,' with an optional port number (i.e., the same as your e-mail address, though many variations of this basic plan are supported, including ways of embedding a standard phone number in an URL). The URL is translated to an IP address (fixed, dynamic, temporary, etc.) through DNS, the generic nameserver system.

"In H.323," explains Schulzrinne, "there is very much a vertical integration notion present. It specifies everything from the codec for the media down to how you carry the packets in RTP, because part of the specification is to describe the content of the data stream. In the IETF [the standards body promoting SIP], we've taken much more of a Lego-like approach, much more horizontal. What we've tried to provide are building blocks, which fit together with a number of different Internet protocols, so that we can use a common URL for naming, we can use MIME for describing content, etc."

Rosenberg emphasizes the point: "We didn't go and define our own type of addresses because we saw that the Internet already had address formats, URLs, and we figured that people are probably going to want to throw together URLs of different types, as they have elsewhere on the Internet. And without even really considering the implications of that decision, the service possibilities it has enabled have been huge. For example, with SIP, it's just as easy to transfer

This year,
do something NOBLE
for your call center...

Get to know NOBLE SYSTEMS!
888 - 8 - NOBLE - 8

Predictive, Outbound, Inbound, ACD, ANI, DNIS, Blended, Digital Recording, Relational Database Real-Time Report Generation, Scripting, List Management, Internal/External Call Transfer, Agent-Level Scheduled Ca Back, Audio & Data Monitoring, Remote Access, Training, Support, Scalable, Modular, Integrated ... Truly Customize

NOBLE SYSTEMS
CUSTOMIZED CALL CENTER AUTOMATION
Suite 550, 4151 Ashford Dunwoody Road
Atlanta, Georgia 30319-1462
Phone 404-851-1331 · Fax 404-851-1421
www.noblesys.com email: info@noblesys.com

Dialogic · SCO · Hewlett Packard · INFORMIX SOLUTIONS ALLIANCE

CTI EXPO · CTI · Product Call Center · CTI EXPO

Enter 91 on card, or at www.computertelephony.com/productinfo

someone to another phone as it is to transfer them to a web page or any other application that accepts URLs — even ones like instant messaging, which didn't exist when we wrote the spec."

The mechanism for doing this magic doesn't even belong to SIP, per se — it just falls out of the decision to use standard DTDs and URLs to manage telephony. In fact, SIP goes well beyond this point in cleaving to broad-based, general purpose standards. Its message codes, for example — relatively few in number — map to HTTP's "first-digit-most-significant" decimal sequence. So (as any web programmer will appreciate) if you get a SIP message code somewhere in the 400's, it means you're doing something wrong. A message in the 500's means the far-end server has crashed.

When a web-oriented programmer looks at SIP, therefore, there's an immediate sense of the familiar. As the programmer digs deeper, there's an even-more-reassuring feeling that SIP does what it does — register endpoints, transmute and pass on information, set up sessions on dynamically-allocated ports, and let endpoints negotiate protocol and codec details — in a minimal, practical way. If IP endpoint addresses are known, a single message exchange suffices to set up a point-to-point call, including real-time protocol and codec determination and dynamic port allocation on each side. H.323v2 can only approach such economy when its as-yet-unstandardized 'fastStart' call setup option is used.

To support mobility and higher-order applications, SIP defines several "useful entities" (read: simple pieces of software that sit on a well-known port) that help manage calls in different ways: registrars, which maintain a map of "what IP address a given user is at, right now"; proxies, which can act as transcoders, auto-responders, and forwarding agents; and redirect servers, which perform a subset of forwarding functions. These helpers can be set up to work around all the common problems of dynamic IP addressing. PC terminals that get turned off, workers that move from place to place, as well as all sorts of higher-order applications.

Again, the general scheme looks familiar to Internet-aware programmers. SIP servers are, of course, complete abstractions — corresponding in no way to "one box per function," except where scale and simplicity mandate this solution. In many cases, a single box will house several — or all — of the discrete functionaries; just as a small-office server may today house a DHCP, a DNS, SMTP/POP3 e-mail servers, an HTTP server, and other elements. As with other Net services, the segregation of SIP entities exists mostly for the sake of practicality. In a typical office, workers turn off their PCs at the end of the day, but the servers keep running. A SIP proxy, mounted on one of the servers, stays up and keeps answering calls.

The proxy is SIP's most powerful "chunk": A complete proxy can redirect, firewall, transcode, reoriginate. And it can house call agents — yet another abstraction, translating roughly to "a virtual endpoint." But even the light-duty functions of a proxy — equivalent to those of a redirect server and simple enough to be encapsulated even in an endpoint, if desired — are enormously powerful. For example, you can adapt an ACD to use a SIP proxy server as its "switching engine" — the proxy takes INVITE requests from callers, returns a 182 'queued' message, then (when agents are free), sends redirect messages to calling clients — the subsequent connections are made point to point.

SIP's intimate association with Net standards and approaches — its consistent use of abstraction and "necessary and sufficient" simplicity — are enormously beneficial. Not only does SIP integrate with, scale in similar fashion to, and otherwise map itself to the Net's most important drivers, but it also establishes telephony as part of a continuum of Net media options — readily accessible to Net programmers, and eventually (through widespread use of SIP CPL — call processing language — and other tools now in the pipeline) to rank-and-file webmonkeys, as well

WHY SIP MAY WIN

Perhaps the most powerful aspect of SIP is again an abstraction. Unlike H.323, which specifies everything but the color of the knobs and dials, SIP doesn't specify anything it doesn't have to. It's just a simple toolkit, atop which smart clients and applications can be built. Ultimately, it means freedom for the enterprise, carriers — the whole telecom ecology. It means enormous variation in how services are deployed, and in what telephony looks like.

For example, SIP is central to several carriers' plans to deploy IP Centrex in tandem with basic Net access, e-mail, domain hosting, DNS, firewall, and other 'business package' services. Plug a hardware SIP phone into a LAN outlet, input your e-mail address and you're done: The phone grabs an IP address from a local DHCP, sends a multicast registration request to the carrier's registrar, and is ready to make and take calls. Once information about colleagues (e.g., their IP addresses) is absorbed by the client, it requires no help from carrier servers to perform most of the functions of a PBX: i.e., it can "extension dial" point to point, put callers on hold, transfer (just send a redirect and have the caller's client reoriginate), etc. Outbound (i.e., outside the enterprise) calls bounce off the carrier's DNS, then out into the cloud (or carrier-maintained gateways). Inbound calls (coming across the Net or through gateways) hit the carrier's proxy momentarily, and get redirected to endpoints. Unless the carrier wants to host messaging, conference bridging, ACD, or other sophisticated services (and many will) basic feature service (except for that annoying gateway maintenance) becomes a "no overhead" proposition.

By the same token, some may choose to house SIP-server and application smarts on-premise — much as today, firms with serious e-commerce ambitions may elect to babysit a rack of web servers. Ultimately, however, this may end up being more a matter of taste than necessity — it's hard to imagine a SIP app that would mandate installing CPE. In fact, we expect SIP to be a

major enabler in the global drive to eliminate or minimize telephony premise equipment. If baby-sitting racks of SIP servers is an issue for anyone, it's going to be third-party ASPs — who'll spring up to OEM-host SIP services to carriers, or rent them directly to customers.

The market for SIP-enabled services will be rich. And — so Schulzrinne, Rosenberg and others predict — it will comprise both classically mass-market and true vertical-market applications, just like the web does, today. Vertical apps will emerge — somewhat ironically — as the result of SIP's horizontal orientation.

"I think, fundamentally, the success of the Internet is all about taking vertical pieces and breaking them into horizontal components," says Rosenberg. "The reason why the Internet succeeded in a lot of cases, where BBSs and other online services had failed, is because the Internet immediately separated out transport from services, while the others tried to integrate access, transport, and services."

"Now we're starting to see Internet telephony following a similar evolution — we've already broken up the telephony gateway, for example, into a softswitch and a media gateway. But at this point, it's still a vertically integrated market. For the evolution to continue, we're going to have to break up the model into even more pieces, so that one user's services can reside in any number of different places in the network, depending on what they are."

On the Internet, this idea is a given. "An ISP doesn't build or own all the web services its users access. It lets other people build web services that are customized for a particular group of people, because that's some other person's expertise," Rosenberg adds. For telecom, however, this proposition involves some major paradigm shifting.

"There used to be a kind of black art," Schulzrinne quips, "where you had to undergo rituals and have your head shaved appropriately before you were allowed to program an SS7 service. It's not something you could just learn in school. But what we're doing is making it possible for people who have a similar skill set to web page designers, who know some standard scripting languages, to develop services that are either customized for their own organization, or target some vertical market."

As Rosenberg points out, "The Internet is all about access to tools. It was because some random yahoo could sit down and say, "Gee, this is a neat idea," and then whip up the service ... that there was so much innovation and so much growth in commerce, all at once. In the voice world, though, I'd have to wait for my telco to go through the three-year cycle of adding a new service, and it would still not be a true vertical-market service. We've never seen vertical-market, specialized voice services, never. But we've seen tons of vertical-market web services deployed in the past few years, and that's where a huge source of value has been. So our mission is to create a horizontal platform that lets anybody create vertical-market services incorporating voice."

ARE WE DREAMING?

A virtual web of decentralized services. Disparate endpoints communicating with one another through nothing more than their own embedded software. Yahoo! deploying voice applications once controlled by AT&T ... Are we living in a fantasy world? Yet every possible indication that we've seen from the industry in recent months suggests that SIP, and its fundamental implications for communication, are about to hit in a big way.

Part of the proof lies in the wide range of products that are already incorporating SIP at different levels in the network. In terms of infrastructure, dynamicsoft is leading the way by providing SIP proxies and location services as the basis for a horizontally integrated applications platform. The softswitch community has largely embraced SIP as a way of communicating between softswitches and is strongly considering the protocol as a means for tying together softswitches and application servers. At the edges, companies like AudioTalk (now HearMe) and NetSpeak are basing software VoIP clients on SIP, and proving the technology to be usable today, for anyone with a multimedia PC and browser. And PingTel is taking the next logical step (really a leap) at the edge by building the first truly intelligent SIP telephones, and using a fully integrated Java environment that demonstrates the extent of the protocol's proximity to the Internet.

Perhaps most significantly, SIP is garnering a high degree of support from carriers. Level 3 has made recent announcements that describe widespread use of SIP throughout its network. And, at the most recent VON show, MCI WorldCom demonstrated a public test network that incorporated SIP-based products from at least seven different vendors, all interoperating with one another.

The scope of SIP-based products and services is likely to grow immensely over the next few months and years. As it does, however, we expect to see the protocol less emphasized rather than more — just as one doesn't necessarily emphasize the use of HTTP in an Internet application. Already, a growing group of players are approaching it from the right direction. "What we've found," says Rosenberg, "is that all sorts of vendors and service providers have particular applications that they want to get done. When they try to figure out how, the find they have a choice of protocols, none of which has already defined the necessary feature set, but which can serve as a platform to build upon. Increasingly, we're finding they want to build on SIP. As a result, we're seeing a lot of vendors defining extensions, and doing things that we hadn't originally conceived of SIP to do; but, then again, that was sort of the whole idea, now, wasn't it?"

3COM

3Com (Santa Clara, CA — 408-326-5000, www.3com.com) has come out strong in support of SIP, and has been active in promoting its acceptance for quite some time. Initial implementations have largely centered around using SIP as a way of interfacing between a Palm Pilot and a SIP-based telephone, which the company has demonstrated at trade shows in recent months. The Palm integration, however,

while holding undeniable sex appeal, as well as the potential for some truly useful applications (dialing directly out of your Palm address book, as a basic example), is really a way of directing attention toward a larger 3Com project — namely the development of an IP Centrex solution.

Having just launched its first beta trials, and with general availability expected this summer, the product is already fairly close to completion. Essentially, the system consists of a series of SPARC servers that will run Centrex-like features, and communicate with client devices over IP, through a SIP server. In addition to the call control and applications software, 3Com has developed the SIP server itself, as well as a line of SIP phones to reside at the customer premise. Nevertheless, the company plans to use fully open interfaces at every level, so that any component of the system — phones, servers, applications — could be replaced or complemented by a standards-based product from a third-party. This open architecture differs from 3Com's enterprise LAN-PBX offering, the NBX-100, which uses a proprietary protocol over Ethernet to interface with phones. Although 3Com eventually plans to migrate the NBX to an open IP protocol, they are first and foremost looking at SIP as a wide area protocol, and are in this respect in line with the thinking of many other vendors. Ikhlaq Suhu, 3Com's VP of Internet communications — network systems business unit, points to SIP's inherent scalability, reliability, and simplicity — all of which are related to the fact that the protocol defines a peer-to-peer, "stateless" call model — as its main advantages for use in networks that extend beyond the local area.

3Com's SIP initiatives include integrating Palm applications with its popular, Ethernet-based NBX-100 phones.

In addition to participating in and hosting SIP bake-offs, 3Com is currently in an IETF proposal to specify an open standard for service provisioning and authentication in SIP-based application server architectures like its own. The company has also submitted a draft to the IETF that defines a standard method for passing SIP messages between a Palm device and a phone. By standardizing its own de facto method for this type of application, 3Com plans to enable developers of Palm apps, as well as vendors of other Palm OS products and other SIP phones to achieve the same type of integration.